

## AMENDMENTS TO THE CLAIMS

The following listing of claims will replace all prior versions and listings of claims in the application:

1. (Previously Canceled)
2. (Previously Presented): A processor comprising:
  - an execution unit to execute instructions;
  - a replay system coupled to the execution unit to replay instructions which have not executed properly, the replay system comprising:
    - a checker to determine whether each instruction has executed properly;
    - a replay queue coupled to the checker to temporarily store one or more instructions for replay;
    - a replay loop to route an instruction which executed improperly to an execution unit for replay; and
    - a replay queue loading controller to determine whether to load an improperly executed instruction to the replay loop or into the replay queue.
3. (Original): The processor of claim 2 and further comprising:
  - a scheduler to output instructions; and
  - a multiplexer or selection mechanism having a first input coupled to the scheduler, a second input coupled to the replay loop and a third input coupled to an output of the replay queue.

4. (Previously Presented): A processor comprising:  
an execution unit to execute instructions;  
a replay system coupled to the execution unit to replay instructions which have not executed properly, the replay system comprising:  
a checker to determine whether each instruction has executed properly; and  
a replay queue coupled to the checker to temporarily store one or more instructions for replay, wherein said replay queue comprises a replay queue coupled to the checker to temporarily store one or more long latency instructions until the instruction is ready for execution.

5. (Previously Presented): A processor comprising:  
an execution unit to execute instructions;  
a replay system coupled to the execution unit to replay instructions which have not executed properly, the replay system comprising:  
a checker to determine whether each instruction has executed properly; and  
a replay queue coupled to the checker to temporarily store one or more instructions for replay, wherein the replay queue comprises a replay queue coupled to the checker to temporarily store an instruction that is not ready to execute properly, the instruction being unloaded from the replay queue when the instruction is ready to execute properly.

6. (Previously Presented) A processor comprising:  
an execution unit to execute instructions;  
a replay system coupled to the execution unit to replay instructions which have not executed properly, the replay system comprising:

a checker to determine whether each instruction has executed properly; and  
a replay queue coupled to the checker to temporarily store one or more  
instructions for replay, wherein the replay queue comprises a replay queue coupled to the  
checker to temporarily store an instruction in which source data must be retrieved from  
an external memory device, the instruction being unloaded from the replay queue when  
the source data for the instruction returns from the external memory device.

7. (Previously Presented): The processor of claim 2 wherein said execution unit is a  
memory load unit, the processor further comprising:

a first level cache system coupled to the memory load unit;  
a second level cache system coupled to the first level cache system; and  
wherein the memory load unit performs a data request to external memory if there  
is a miss on both the first level and second level cache systems.

8. (Original): The processor of claim 7 wherein a load instruction will be loaded  
into the replay queue when there is a miss on both the first level and second level cache  
systems, and the load instruction is unloaded from the replay queue for re-execution  
when the data for the instruction returns from the external memory.

9. (Original): A processor comprising:  
a multiplexer having an output;  
a scheduler coupled to a first input of the multiplexer;  
an execution unit coupled to an output of the multiplexer;

a checker coupled to the output of the multiplexer to determine whether an instruction has executed properly;

a replay queue to temporarily store instructions, an output of the replay queue coupled to a second input of the multiplexer; and

a controller coupled to the checker to determine when to load an instruction into the replay queue and to determine when to unload the replay queue.

10. (Original): The processor of claim 9 and further comprising a staging section coupled between the checker and a third input to the multiplexer to provide a replay loop, the controller controlling the multiplexer to select either the output of the scheduler, the replay loop or the output of the replay queue.

11. (Original): The processor of claim 9 wherein the controller loads an instruction into the replay queue when the instruction is not ready to execute properly, and unloads the instruction from the replay queue when the instruction is ready to execute properly.

12. (Original): The processor of claim 9 wherein the controller determines when to unload the replay queue based on a data return signal.

13—20. (Previously Canceled)

21. (New): The processor of claim 4, wherein if said long latency instruction is ready for execution, the long latency instruction is unloaded from the replay queue.

22. (New): The processor of claim 5, wherein a controller determines whether to unload the replay queue based on a data return signal.
23. (New): The processor of claim 6 wherein a controller determines whether to unload the replay queue based on a data return signal.